1. Azkaban 概述

1.1. 为什么需要工作流调度器

```
1、一个完整的数据分析系统通常都是由大量任务单元组成：shell脚本程序，java程序，mapreduce程序、hive脚本等
2、各任务单元之间存在时间先后及前后依赖关系
3、为了很好地组织起这样的复杂执行计划，需要一个工作流调度系统来调度执行
```

例如，我们可能有这样一个需求，某个业务系统每天产生 20G 原始数据，我们每天都要对其进行处理，处理步骤如下所示：

```
1、通过 Hadoop 先将原始数据同步到 HDFS 上；
2、借助 MapReduce 计算框架对原始数据进行清洗转换，生成的数据以分区表的形式存储到多张 Hive 表中；
3、需要对 Hive 中多个表的数据进行 Join 处理，得到一个明细数据 Hive 大表；
4、将明细数据进行各种统计分析，得到结果报表信息；
5、需要将统计分析得到的结果数据同步到业务系统中，供业务调用使用。
```

1.2. 工作流调度实现方式

简单的任务调度：

```
直接使用 Linux 的 Crontab 来定义
```

复杂的任务调度：

```
自己开发调度平台或使用现成的开源调度系统，比如 oozie、azkaban, airflow 等
```

1.3. 常见工作流调度系统

在 Hadoop 领域，常见的工作流调度器有 Oozie，Azkaban，Cascading，Hamake 等

1.4. 各种调度工具对比

下面的表格对上述四种 Hadoop 工作流调度器的关键特性进行了比较，尽管这些工作流调度器能够解决的需求场景基本一致，但在设计理念，目标用户，应用场景等方面还是存在显著的区别，在做技术选型的时候，可以提供参考

特性	Hamake	Oozie	Azkaban	Cascading
工作流描述语言	XML	XML(xPDL based)	text file with key/value pairs	Java API
依赖机制	data-driven	explicit/data-driven	explicit	explicit
是否要web容器	No	Yes	Yes	No
进度跟踪	console/log messages	web page	web page	Java API
HadoopJob调度支持	no	yes	yes	yes
运行模式	command line utility	daemon	daemon	API
Pig支持	yes	yes	yes	yes
事件通知	no	no	no	yes
需要安装	no	yes	yes	no
支持的hadoop版本	0.18+	0.20+	currently unknown	0.18+
重试支持	no	workflownode evel	yes	yes
运行任意命令	yes	yes	yes	yes
Amazon EMR支持	yes	no	currently unknown	yes

1.5. Azkaban与Oozie对比

对市面上最流行的两种调度器，给出以下详细对比，以供技术选型参考。总体来说，ooize 相比 azkaban 是一个重量级的任务调度系统，功能全面，但配置使用也更复杂。如果可以不在意某些功能的缺失，轻量级调度器 azkaban 是很不错的候选对象。

详情如下：

功能

```
两者均可以调度MapReduce，Hive，Java，脚本工作流任务等
两者均可以定时执行和间隔执行工作流任务
```

工作流定义

```
Azkaban使用Properties文件定义工作流
Oozie使用XML文件定义工作流
```

工作流传参

```
Azkaban支持直接传参，例如${input}
Oozie支持参数和EL表达式，例如${fs:dirSize(myInputDir)} strust2(ONGL)
```

定时执行

```
Azkaban的定时执行任务是基于时间的
Oozie的定时执行任务基于时间和输入数据
```

资源管理

```
Azkaban有较严格的权限控制，如用户对工作流进行读/写/执行等操作
Oozie暂无严格的权限控制
```

工作流执行

```
Azkaban有两种运行模式，分别是solo server mode(executor server和web server部署在同一台节点)和multi server mode(executor server和web server可以部署在不同节点)
Oozie作为工作流服务器运行，支持多用户和多工作流
```

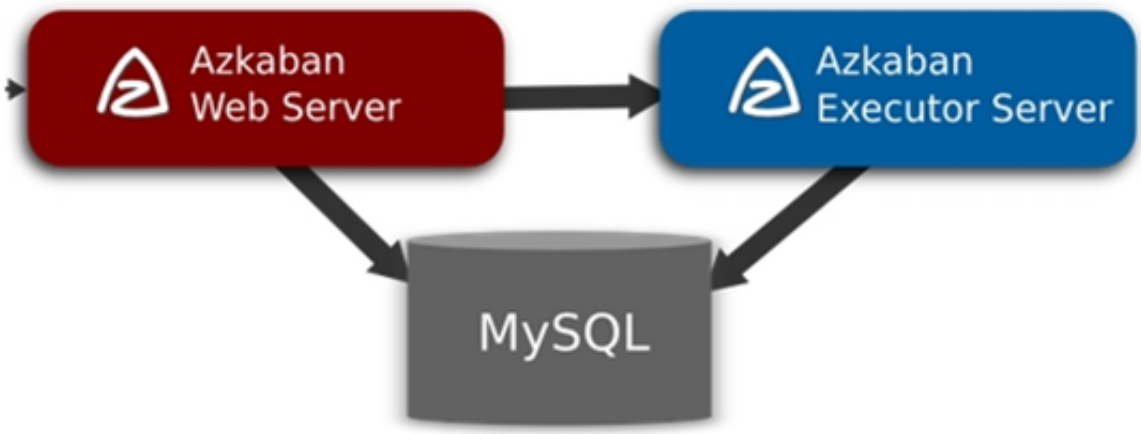
工作流管理

```
Azkaban支持浏览器以及ajax方式操作工作流
Oozie支持命令行、HTTP REST、Java API、浏览器操作工作流
```

1.6. Azkaban介绍

Azkaban 是由 Linkedin 开源的一个批量工作流任务调度器。用于在一个工作流内以一个特定的顺序运行一组工作和流程。Azkaban 定义了一种 Key-Value 形式的 Job 配置文件 (properties) 格式来建立任务之间的依赖关系，并提供一个易于使用的web用户界面维护和跟踪你的工作流。它有三个重要组件：

> 关系数据库（目前仅支持MySQL）
> web管理服务器 - AzkabanWebServer
> 执行服务器 - AzkabanExecutorServer



Azkaban 使用 MySQL 来存储它的状态信息，Azkaban Executor Server 和 Azkaban Web Server 均使用到了MySQL 数据库。

它有如下功能特点：

```
web用户界面
方便上传工作流
方便设置任务之间的关系
调度工作流
认证/授权(权限的工作)
能够杀死并重新启动工作流
模块化和可插拔的插件机制
项目工作区
工作流和任务的日志记录和审计
```

2. Azkaban安装部署

2.1. 准备工作

Azkaban Web服务器：azkaban-web-server-2.5.0.tar.gz
Azkaban Excutor 执行服务器：azkaban-executor-server-2.5.0.tar.gz
Azkaban 初始化脚本文件：azkaban-sql-script-2.5.0.tar.gz

下载地址：http://azkaban.github.io/downloads.html

2.2. 安装说明

将安装文件上传到集群，最好上传到安装 hive、sqoop 的机器上，方便命令的执行。并最好同一存放在 apps 目录下，用于存放源安装文件。新建 azkaban 目录，用于存放 azkaban 运行程序

```
mkdir ~/apps/azkaban
```

2.3. 安装 Azkaban Web 服务器

```
tar -zxvf ~/soft/azkaban-web-server-2.5.0.tar.gz -C ~/apps/azkaban/
```

2.4. 安装 Azkaban Excutor 服务器

```
tar -zxvf ~/soft/azkaban-executor-server-2.5.0.tar.gz -C ~/apps/azkaban/
```

2.5. 安装 Azkaban 脚本导入

注意该操作要在你的 MySQL 服务节点上执行：

```
tar -zxvf ~/soft/azkaban-sql-script-2.5.0.tar.gz -C ~/apps/azkaba
```

或者当前节点解压了以后，把要用的SQL脚本文件：create-all-sql-2.5.0.sql 发送到对应的MySQL节点也可以。

```
scp /home/bigdata/apps/azkaban/azkaban-2.5.0/create-all-sql-2.5.0.sql
bigdata02:~
```

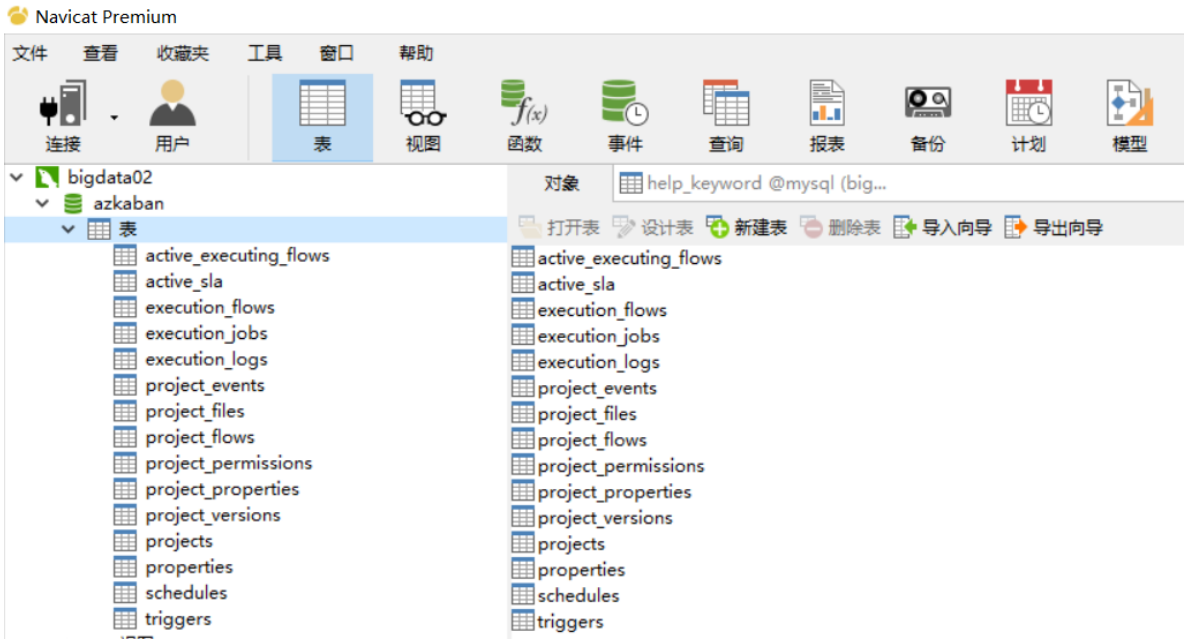
进入 MySQL：

```
mysql> create database azkaban;
Query OK, 1 row affected (0.01 sec)

mysql> use azkaban;
Database changed

mysql> source /home/bigdata/create-all-sql-2.5.0.sql;
```

看结果：



2.6. 创建SSL配置

参考地址: http://docs.codehaus.org/display/JETTY/How+to+configure+SSL

最好是在 azkaban 目录下：

执行命令：

```
keytool -keystore keystore -alias jetty -genkey -keyalg RSA
```

运行此命令后，会提示输入当前生成 keystore 的密码及相应信息，输入密码请劳记,信息如下：

```
[bigdata@bigdata05 azkaban]# keytool -keystore keystore -alias jetty -genkey -
keyalg RSA
```

```
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:  CN
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=CN correct?
  [no]:  y

Enter key password for <jetty>
        (RETURN if same as keystore password):

[bigdata@bigdata05 azkaban]# ll
total 16
drwxr-xr-x. 7 root root 4096 Nov 21 01:53 azkaban-executor-2.5.0
drwxr-xr-x. 2 root root 4096 Nov 21 01:53 azkaban-script-2.5.0
drwxr-xr-x. 8 root root 4096 Nov 21 01:52 azkaban-web-2.5.0
-rw-r--r--. 1 root root 2232 Nov 21 02:06 keystore
```

完成上述工作后，将在当前目录生成 keystore 证书文件,将keystore 拷贝到 azkaban web服务器根目录中.如：

```
cp keystore azkaban-web-2.5.0
```

2.7. 修改配置文件

注：先配置好服务器节点上的时区

1、先生成时区配置文件 Asia/Shanghai，用交互式命令 tzselect 即可
2、拷贝该时区文件，覆盖系统本地时区配置

```
sudo cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

azkaban web服务器配置

进入 azkaban web 服务器安装目录 conf 目录

```
cd ~/apps/azkaban/azkaban-web-2.5.0/conf/
```

修改 azkaban.properties 文件

```
vim azkaban.properties
```

内容说明如下：

```
#Azkaban Personalization Settings
azkaban.name=MyTestAzkaban                              #服务器UI名称,用于服务器上方显示的名字
azkaban.label=My Local Azkaban                          #描述
azkaban.color=#FF3601                                   #UI颜色
azkaban.default.servlet.path=/index
web.resource.dir=/home/bigdata/apps/azkaban/azkaban-web-2.5.0/web/      #默认根web
目录
default.timezone.id=Asia/Shanghai                       #默认时区,已改为亚洲/上海 默认为美国

#Azkaban UserManager class
user.manager.class=azkaban.user.XmlUserManager          #用户权限管理默认类
user.manager.xml.file=/home/bigdata/apps/azkaban/azkaban-web-2.5.0/conf/azkaban-
users.xml
                                                        #用户配置,具体配置参加下文

#Loader for projects                # global配置文件所在位置
executor.global.properties=/home/bigdata/apps/azkaban/azkaban-executor-
2.5.0/conf/global.properties
azkaban.project.dir=projects

database.type=mysql                 #数据库类型
mysql.port=3306                     #端口号
mysql.host=bigdata02                #数据库连接IP
mysql.database=azkaban              #数据库实例名
mysql.user=root                     #数据库用户名
mysql.password=QWer_1234            #数据库密码
mysql.numconnections=100            #最大连接数

# Velocity dev mode
velocity.dev.mode=false
# Jetty服务器属性.
jetty.maxThreads=25                 #最大线程数
jetty.ssl.port=8443                 #Jetty SSL端口
jetty.port=8081                     #Jetty端口
jetty.keystore=/home/bigdata/apps/azkaban/azkaban-web-2.5.0/keystore        #SSL
文件名
jetty.password=bigdata                          #SSL文件密码
jetty.keypassword=bigdata                       #Jetty主密码 与 keystore文件相同
jetty.truststore=/home/bigdata/apps/azkaban/azkaban-web-2.5.0/keystore   #SSL文件
名
jetty.trustpassword=bigdata                                                 #SSL文件
密码

# 执行服务器属性
executor.port=12321                             #执行服务器端口

# 邮件设置(可选项)
mail.sender=xxxxxxxx@163.com                    #发送邮箱
mail.host=smtp.163.com                          #发送邮箱smtp地址
mail.user=xxxxxxxx                              #发送邮件时显示的名称
mail.password=**********                        #邮箱密码
job.failure.email=xxxxxxxx@163.com              #任务失败时发送邮件的地址
job.success.email=xxxxxxxx@163.com              #任务成功时发送邮件的地址
lockdown.create.projects=false                  #
cache.directory=cache                           #缓存目录
```

用户配置

进入 azkaban web 服务器 conf 目录，修改 azkaban-users.xml

```
vim azkaban-users.xml
```

增加 管理员用户

```
<azkaban-users>
    <user username="azkaban" password="azkaban" roles="admin" groups="azkaban"
/>
    <user username="metrics" password="metrics" roles="metrics"/>
    <user username="admin" password="admin" roles="admin,metrics" />
    <role name="admin" permissions="ADMIN" />
    <role name="metrics" permissions="METRICS"/>
</azkaban-users>
```

azkaban 执行服务器 executor 配置

进入执行服务器安装目录 conf，修改 azkaban.properties

```
[bigdata@bigdata05 ~]# cd ~/apps/azkaban/azkaban-executor-2.5.0/conf/
```

修改文件内容：

```
vi azkaban.properties
```

```
#Azkaban
default.timezone.id=Asia/Shanghai                                    #时区

# Azkaban JobTypes 插件配置，插件所在位置
azkaban.jobtype.plugin.dir=/home/bigdata/apps/azkaban/azkaban-executor-
2.5.0/plugins/jobtypes

#Loader for projects
executor.global.properties=/home/bigdata/apps/azkaban/azkaban-executor-
2.5.0/conf/global.properties
azkaban.project.dir=projects

#数据库设置
database.type=mysql                                  #数据库类型(目前只支持mysql)
mysql.port=3306                                      #数据库端口号
mysql.host=bigdata03                                 #数据库IP地址
mysql.database=azkaban                               #数据库实例名
mysql.user=root                                      #数据库用户名
mysql.password=root                                  #数据库密码
mysql.numconnections=100                             #最大连接数

# 执行服务器配置
executor.maxThreads=50                               #最大线程数
executor.port=12321                                  #端口号(如修改,请与web服务中一致)
executor.flow.threads=30                             #线程数
```

2.8. 配置环境变量

修改环境变量：

```
vim ~/.bashrc
```

```
export AZKABAN_WEB_HOME=/home/bigdata/apps/azkaban/azkaban-web-2.5.0
export AZKABAN_EXE_HOME=/home/bigdata/apps/azkaban/azkaban-executor-2.5.0
export PATH=$PATH:$AZKABAN_WEB_HOME/bin:$AZKABAN_EXE_HOME/bin
```

执行命令使之生效

```
source ~/.bashrc
```

2.9. 启动

2.9.1. 启动 web server

直接前台启动：

```
azkaban-web-start.sh
```

或者使用非挂起的方式启动到后台：

```
nohup azkaban-web-start.sh 1>/home/bigdata/logs/azwebstd.out
2>/home/bigdata/logs/azweberr.out &
```

2.9.2. 启动 Executor Server

直接前台启动：

```
azkaban-executor-start.sh
```

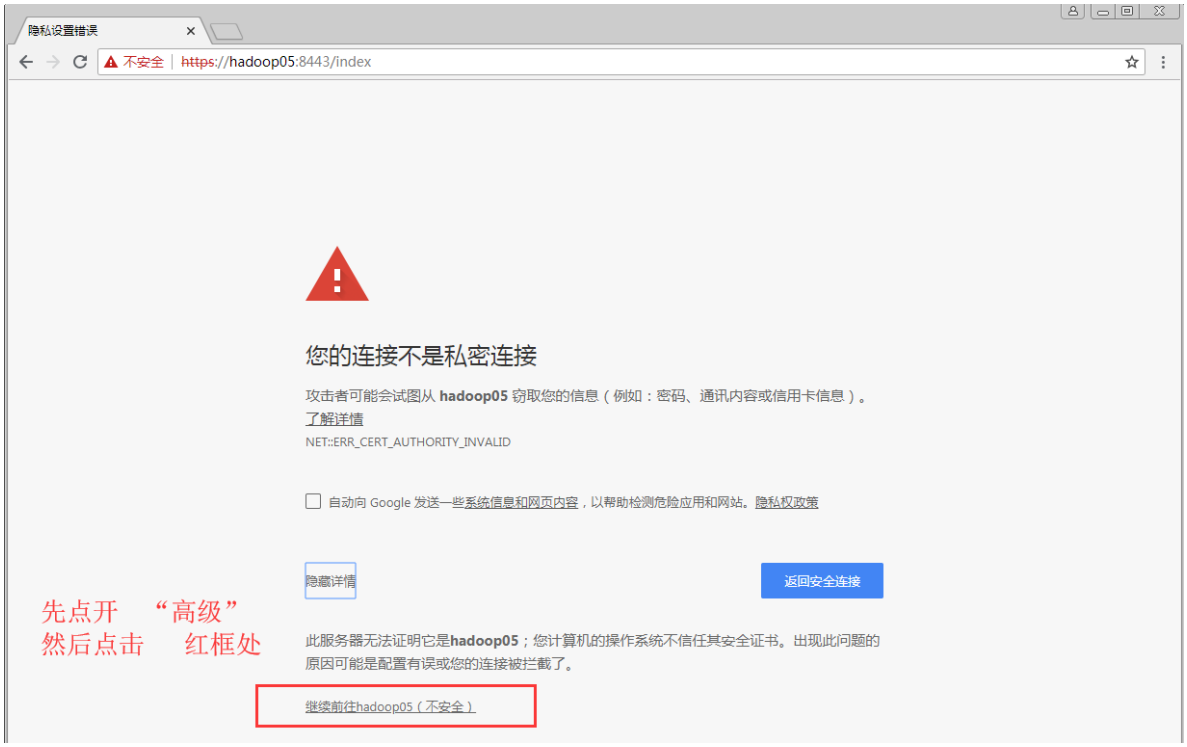
或者使用非挂起的方式启动到后台：

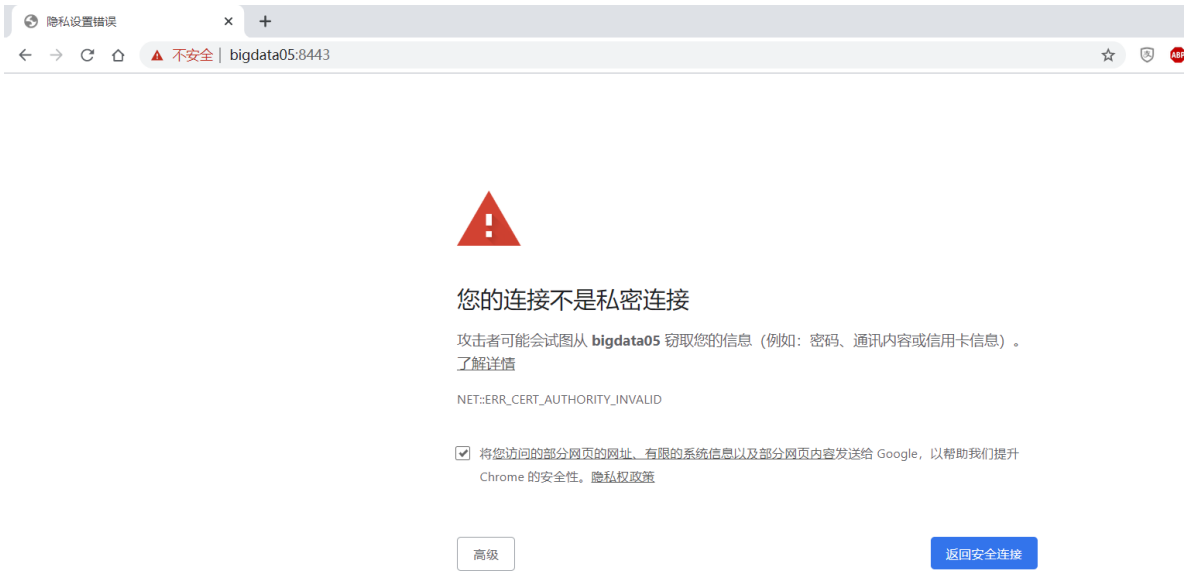
```
nohup azkaban-executor-start.sh 1>/home/bigdata/logs/azexstd.out
2>/home/bigdata/logs/azexerr.out &
```

2.9.3. 访问 web ui

启动完成后，在浏览器(建议使用谷歌浏览器)中输入 https://IP:8443，即可访问azkaban服务了。在登录中输入刚才新的户用名及密码,点击 login。能进入到如下画面证明安装成功。

注意地址是：https://bigdata05:8443/





3. Azkaban实战演示

3.1. Command 类型单一 job 示例

1、创建job描述文件

```
vim command.job
```

```
#command.job
type=command
command=echo 'hello'
```

2、将job资源文件打包

```
zip command.job
```

3、 通过 azkaban web 管理平台创建 project 并上传压缩包

4、调度执行

两种执行方式：

> 第一种：execute now
> 第二种：schedule

3.2. Command 类型多 job 工作流 flow

1、 创建有依赖关系的多个job描述

第一个 job：stepone.job

```
# stepone.job
type=command
command=echo stepone
```

第二个 job：steptwo.job依赖stepone.job

```
# steptwo.job
type=command
dependencies=stepone
command=echo steptwo
```

2、 将所有资源打成一个zip包

3、 创建azkaban任务，然后上传zip包资料，启动执行

3.3. 操作 HDFS 任务

1、 创建job描述文件

```
# hdfs.job
type=command
command=/app/bigdata/apps/hadoop-2.7.7/bin/hadoop fs -mkdir -p /hello/azkaban
```

2、 将job资源文件打成zip包

3、 创建project并上传zip包

4、 启动执行

3.4. 操作 MapReduce 任务

1、 创建job描述文件

```
# mapreduce.job
type=command
command=/home/bigdata/apps/hadoop-2.7.7/bin/hadoop  jar hadoop-mapreduce-
examples-2.7.7.jar wordcount /wordcount/input /wordcount/azout
```

2、 将job资源文件打成zip包

3、 创建project并上传zip包

4、 启动执行

3.5. Hive 脚本任务

1、Hive脚本如下：**hivetest.sql**

```
create database if not exists azkaban_test;
use azkaban_test;
drop table if exists aztest;
create table aztest(id int, name string, sex string, age int, deparment string)
row format delimited fields terminated by ',';
load data local inpath '/home/bigdata/student.txt' into table aztest;
create table az_result as select * from aztest;
insert overwrite directory '/aztest/hiveoutput' select count(1) from az_result;
```

2、 创建 job 描述文件和 hive 脚本

```
# hivef.job
type=command
command=/root/apps/apache-hive-2.3.6-bin/bin/hive -f 'testhive.sql'
```

3、 将job资源文件打成zip包

4、 创建project并上传zip包

5、 启动执行

3.6. Sqoop 数据迁移任务

当做作业完成

3.7. Azkaban 调度完成 Python 脚本任务

等学习 python 之后，自己尝试摸索